Recursive $\ell_{1,\infty}$ Group lasso

Yilun Chen, *Student Member, IEEE,* and Alfred O. Hero, III, *Fellow, IEEE*

***Abstract*—We introduce a recursive adaptive group lasso algorithm for real-time penalized least squares prediction that produces a time sequence of optimal sparse predictor coefficient vectors. At each time index the proposed algorithm computes an exact update of the optimal $\ell_{1,\infty}$-penalized recursive least squares (RLS) predictor. Each update minimizes a convex but non-differentiable function optimization problem. We develop an online homotopy method to reduce the computational complexity. Numerical simulations demonstrate that the proposed algorithm outperforms the ℓ_1 regularized RLS algorithm for a group sparse system identification problem and has lower implementation complexity than direct group lasso solvers.**

***Index Terms*—RLS, group sparsity, mixed norm, homotopy, group lasso, system identification**

I. Introduction

Recursive Least Squares (RLS) is a widely used method for adaptive filtering and prediction in signal processing and related fields. Its applications include: acoustic echo cancelation; wireless channel equalization; interference cancelation and data streaming predictors. In these applications a measurement stream is recursively fitted to a linear model, described by the coefficients of an FIR prediction filter, in such a way to minimize a weighted average of squared residual prediction errors. Compared to other adaptive filtering algorithms such as Least Mean Square (LMS) filters, RLS is popular because of its fast convergence and low steady-state error.

In many applications it is natural to constrain the predictor coefficients to be sparse. In such cases the adaptive FIR prediction filter is a sparse system: only a few of the impulse response coefficients are non-zero. Sparse systems can be divided into general sparse systems and group sparse systems [1], [2]. Unlike a general sparse system, whose impulse response can have arbitrary sparse structure, a group sparse system has impulse response composed of a few distinct clusters of non-zero coefficients. Examples of group sparse systems include specular multipath acoustic and wireless channels [3], [4] and compressive spectrum sensing of narrowband sources [5].

The exploitation of sparsity to improve prediction performance has attracted considerable interest. For general sparse systems, the ℓ_1 norm has been recognized as an effective promotor of sparsity [6], [7]. In particular, ℓ_1 regularized LMS [2], [8] and RLS [9], [10] algorithms have been proposed for for sparsification of adaptive filters. For group sparse systems, mixed norms such as the $\ell_{1,2}$ norm and the $\ell_{1,\infty}$ norm have been applied to promote group-level sparsity in statistical regression [11]–[13], commonly referred to as the group lasso, and sparse signal recovery in signal processing and communications [1], [14]. In [2], the authors proposed a family of LMS filters with convex regularizers. As a specific scenario, they considered group sparse LMS filters with $\ell_{1,2}$-type regularizers and empirically demonstrated the superior performance of $\ell_{1,2}$-regularized LMS over the ℓ_1-regularized LMS for identifying unkonwn group-sparse channels.

In this paper, we develop group sparse RLS algorithms for real-time system identification. Specifically, we consider RLS penalized by the $\ell_{1,\infty}$ norm to promote group sparsity, which we call the recursive $\ell_{1,\infty}$ group lasso. The algorithm is based on the homotopy approach to solving the lasso problem and is a nontrivial extension of [15]–[17] to group sparse structure. Both the $l_{1,2}$ and $l_{1,\infty}$ regularizer have been widely adopted to enforce group sparse structure in regression and other problems. While, as shown in [18], in some cases the $l_{1,2}$ is more effective in enforcing group sparsity, we adopt the $l_{1,\infty}$ norm due to its more favorable computational properties. As we show in this paper, the $l_{1,\infty}$ regularizer can be very efficiently implemented in the recursive setting of RLS. Our implementation exploits the piecewise linearity of the $l_{1,\infty}$ norm to obtain an exact solution to each iteration of the group sparsity-penalized RLS algorithm using the homotopy approach. Compared to other contemporary $\ell_{1,\infty}$ and $\ell_{1,2}$ group lasso solvers [12], [19], our simulation results demonstrate that our proposed methods attain equivalent or better performance but at lower computational cost for online group sparse RLS problems.

The paper is organized as follows. Section II formulates the problem. In Section III we develop the homotopy based algorithm to solve the recursive $\ell_{1,\infty}$ group lasso in an online recursive manner. Section IV provides numerical simulation results and Section V summarizes our principal conclusions. The proofs of theorems and some details of the proposed algorithm are provided in Appendix.

Notations: In the following, matrices and vectors are denoted by boldface upper case letters and boldface lower case letters, respectively; $(\cdot)^T$ denotes the transpose operator, and $\|\cdot\|_1$ and $\|\cdot\|_\infty$ denote the ℓ_1 and ℓ_∞ norm of a vector, respectively; for a set $\mathcal{A}$, $|\mathcal{A}|$ denotes its cardinality and ϕ denotes the empty set; $\mathbf{x}_\mathcal{A}$ denotes the sub-vector of $\mathbf{x}$ from the index set $\mathcal{A}$ and $\mathbf{R}_{\mathcal{A}\mathcal{B}}$ denotes the sub-matrix of $\mathbf{R}$ formed from the row index set $\mathcal{A}$ and column index set $\mathcal{B}$.

II. Problem formulation

A. Recursive Least Squares

Let $\mathbf{w}$ be a p-dimensional coefficient vector. Let $\mathbf{y}$ be an n-dimensional vector comprised of observations $\{y_j\}_{j=1}^n$. Let

Y. Chen and A. O. Hero are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA. Tel: 1-734-763-0564. Fax: 1-734-763-8041. Emails: {yilun, hero}@umich.edu.

This work was partially supported by AFOSR, grant number FA9550-06-1-0324.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
AUG 2012		**00-00-2012 to 00-00-2012**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Recursive l1,oo Group lasso	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Michigan,Department of Electrical Engineering and Computer Science,1301 Beal Avenue,Ann Arbor,MI,48109**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
IEEE Trans. on Signal Processing, vol 68, no 8, pp. 3978-3987, Aug 2012. Author preprint.

14. ABSTRACT
We introduce a recursive adaptive group lasso algorithm for real-time penalized least squares prediction that produces a time sequence of optimal sparse predictor coefficient vectors. At each time index the proposed algorithm computes an exact update of the optimal ‘1;1-penalized recursive least squares (RLS) predictor. Each update minimizes a convex but nondifferentiable function optimization problem. We develop an online homotopy method to reduce the computational complexity. Numerical simulations demonstrate that the proposed algorithm outperforms the ‘1 regularized RLS algorithm for a group sparse system identification problem and has lower implementation complexity than direct group lasso solvers.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**10**	

$\{\mathbf{x}_j\}_{j=1}^n$ be a sequence of p-dimensional predictor variables. In standard adaptive filtering terminology, y_j, $\mathbf{x}_j$ and $\mathbf{w}$ are the primary signal, the reference signal, and the adaptive filter weights. The RLS algorithm solves the following quadratic minimization problem recursively over time $n = p, p+1, \ldots$:

$$\hat{\mathbf{w}}_n = \arg\min_{\mathbf{w}} \sum_{j=1}^{n} \gamma^{n-j}(y_j - \mathbf{w}^T\mathbf{x}_j)^2, \tag{1}$$

where $\gamma \in (0, 1]$ is the forgetting factor controlling the trade-off between transient and steady-state behaviors.

To serve as a template for the sparse RLS extensions described below we briefly review the RLS update algorithm. Define $\mathbf{R}_n$ and $\mathbf{r}_n$ as

$$\mathbf{R}_n = \sum_{j=1}^{n} \gamma^{n-j}\mathbf{x}_j\mathbf{x}_j^T \tag{2}$$

and

$$\mathbf{r}_n = \sum_{j=1}^{n} \gamma^{n-j}\mathbf{x}_j y_j. \tag{3}$$

The solution $\hat{\mathbf{w}}_n$ to (1) can be then expressed as

$$\hat{\mathbf{w}}_n = \mathbf{R}_n^{-1}\mathbf{r}_n. \tag{4}$$

The matrix $\mathbf{R}_n$ and the vector $\mathbf{r}_n$ are updated as

$$\mathbf{R}_n = \gamma\mathbf{R}_{n-1} + \mathbf{x}_n\mathbf{x}_n^T,$$

and

$$\mathbf{r}_n = \gamma\mathbf{r}_{n-1} + \mathbf{x}_n y_n^T.$$

Applying the Sherman-Morrison-Woodbury formula [20],

$$\mathbf{R}_n^{-1} = \gamma^{-1}\mathbf{R}_{n-1}^{-1} - \gamma^{-1}\alpha_n\mathbf{g}_n\mathbf{g}_n^T, \tag{5}$$

where

$$\mathbf{g}_n = \mathbf{R}_{n-1}^{-1}\mathbf{x}_n \tag{6}$$

and

$$\alpha_n = \frac{1}{\gamma + \mathbf{x}_n^T\mathbf{g}_n}. \tag{7}$$

Substituting (5) into (4), we obtain the weight update [21]

$$\hat{\mathbf{w}}_n = \hat{\mathbf{w}}_{n-1} + \alpha_n\mathbf{g}_n e_n, \tag{8}$$

where

$$e_n = y_n - \hat{\mathbf{w}}_{n-1}^T\mathbf{x}_n. \tag{9}$$

Equations (5)-(9) define the RLS algorithm which has computational complexity of order $\mathcal{O}(p^2)$.

B. Non-recursive $\ell_{1,\infty}$ group lasso

The $\ell_{1,\infty}$ group lasso is a regularized least squares approach which uses the $\ell_{1,\infty}$ mixed norm to promote group-wise sparse pattern on the predictor coefficient vector. The $\ell_{1,\infty}$ norm of a vector $\mathbf{w}$ is defined as

$$\|\mathbf{w}\|_{1,\infty} = \sum_{m=1}^{M} \|\mathbf{w}_{\mathcal{G}_m}\|_\infty,$$

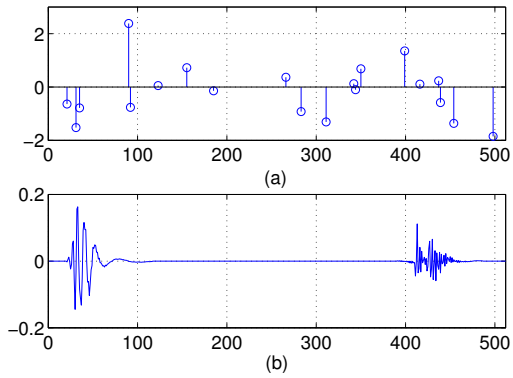

Fig. 1. Examples of (a) a general sparse system and (b) a group-sparse system.

where $\{\mathcal{G}_m\}_{m=1}^M$ is a group partition of the index set $\mathcal{G} = \{1, \ldots, p\}$, *i.e.,*

$$\bigcup_{m=1}^{M} \mathcal{G}_m = \mathcal{G}, \quad \mathcal{G}_m \cap \mathcal{G}_{m'} = \phi \text{ if } m \neq m',$$

and $\mathbf{w}_{\mathcal{G}_m}$ is a sub-vector of $\mathbf{w}$ indexed by $\mathcal{G}_m$. The $\ell_{1,\infty}$ norm is a mixed norm: it encourages correlation among coefficients inside each group via the ℓ_∞ norm within each group and promotes sparsity across each group using the ℓ_1 norm. The mixed norm $\|\mathbf{w}\|_{1,\infty}$ is convex in $\mathbf{w}$ and reduces to $\|\mathbf{w}\|_1$ when each group contains only one coefficient, *i.e.,* $|\mathcal{G}_1| = |\mathcal{G}_2| = \cdots = |\mathcal{G}_M| = 1$.

The $\ell_{1,\infty}$ group lasso solves the following penalized least squares problem:

$$\hat{\mathbf{w}}_n = \arg\min_{\mathbf{w}} \frac{1}{2}\sum_{j=1}^{n} \gamma^{n-j}(y_j - \mathbf{w}^T\mathbf{x}_j)^2 + \lambda\|\mathbf{w}\|_{1,\infty}, \tag{10}$$

where λ is a regularization parameter and as in standard RLS γ controls the trade-off between the convergence rate and steady-state performance. Eq. (10) is a convex problem and can be solved by standard convex optimizers or path tracing algorithms [12]. Direct solution of (10) has computational complexity of $\mathcal{O}(p^3)$.

C. Recursive $\ell_{1,\infty}$ group lasso

In this subsection we obtain a recursive solution for (10) that gives an update $\hat{\mathbf{w}}_n$ from $\hat{\mathbf{w}}_{n-1}$. The approach taken is a group-wise generalization of recent works [15], [16] that uses the homotopy approach to sequentially solve the lasso problem. Using the definitions (2) and (3), the problem (10) is equivalent to

$$\begin{aligned}\hat{\mathbf{w}}_n &= \arg\min_{\mathbf{w}} \frac{1}{2}\mathbf{w}^T\mathbf{R}_n\mathbf{w} - \mathbf{w}^T\mathbf{r}_n + \lambda\|\mathbf{w}\|_{1,\infty} \\ &= \arg\min_{\mathbf{w}} \frac{1}{2}\mathbf{w}^T\left(\gamma\mathbf{R}_{n-1} + \mathbf{x}_n^T\mathbf{x}_n\right)\mathbf{w} \\ &\quad - \mathbf{w}^T(\gamma\mathbf{r}_{n-1} + \mathbf{x}_n y_n) + \lambda\|\mathbf{w}\|_{1,\infty}.\end{aligned} \tag{11}$$

Let $f(\beta,\lambda)$ be the solution to the following parameterized problem

$$f(\beta,\lambda) = \arg\min_{\mathbf{w}} \frac{1}{2}\mathbf{w}^T\left(\gamma\mathbf{R}_{n-1} + \beta\mathbf{x}_n\mathbf{x}_n^T\right)\mathbf{w} - \mathbf{w}^T(\gamma\mathbf{r}_{n-1} + \beta\mathbf{x}_n y_n) + \lambda\|\mathbf{w}\|_{1,\infty} \quad (12)$$

where β is a constant between 0 and 1. $\hat{\mathbf{w}}_n$ and $\hat{\mathbf{w}}_{n-1}$ of problem (11) can be expressed as

$$\hat{\mathbf{w}}_{n-1} = f(0,\gamma\lambda),$$

and

$$\hat{\mathbf{w}}_n = f(1,\lambda).$$

Our proposed method computes $\hat{\mathbf{w}}_n$ from $\hat{\mathbf{w}}_{n-1}$ in the following two steps:

Step 1. Fix $\beta = 0$ and calculate $f(0,\lambda)$ from $f(0,\gamma\lambda)$. This is accomplished by computing the regularization path between $\gamma\lambda$ and λ using homotopy methods introduced for the non-recursive $\ell_{1,\infty}$ group lasso. The solution path is piecewise linear and the algorithm is described in [12].

Step 2. Fix λ and calculate the solution path between $f(0,\lambda)$ and $f(1,\lambda)$. This is the key problem addressed in this paper.

To ease the notations we denote $\mathbf{x}_n$ and y_n by $\mathbf{x}$ and y, respectively, and define the following variables:

$$\mathbf{R}(\beta) = \gamma\mathbf{R}_{n-1} + \beta\mathbf{x}\mathbf{x}^T \quad (13)$$

$$\mathbf{r}(\beta) = \gamma\mathbf{r}_{n-1} + \beta\mathbf{x}y. \quad (14)$$

Problem (12) is then

$$f(\beta,\lambda) = \arg\min_{\mathbf{w}} \frac{1}{2}\mathbf{w}^T\mathbf{R}(\beta)\mathbf{w} - \mathbf{w}^T\mathbf{r}(\beta) + \lambda\|\mathbf{w}\|_{1,\infty}. \quad (15)$$

In Section III we will show how to propagate $f(0,\lambda)$ to $f(1,\lambda)$ using the homotopy approach applied to (15).

III. Online homotopy update

A. Set notation

We begin by introducing a series of set definitions. Figure 2 provides an example. We divide the entire group index set into $\mathcal{P}$ and Q, respectively, where $\mathcal{P}$ contains active groups and $\mathcal{Q}$ is its complement. For each active group $m \in \mathcal{P}$, we partition the group into two parts: the maximal values, with indices $\mathcal{A}_m$, and the rest of the values, with indices $\mathcal{B}_m$:

$$\mathcal{A}_m = \arg\max_{i\in\mathcal{G}_m} |w_i|, m \in \mathcal{P},$$

and

$$\mathcal{B}_m = \mathcal{G}_m - \mathcal{A}_m.$$

The set $\mathcal{A}$ and $\mathcal{B}$ are defined as the union of the $\mathcal{A}_m$ and $\mathcal{B}_m$ sets, respectively:

$$\mathcal{A} = \bigcup_{m\in\mathcal{P}} \mathcal{A}_m, \quad \mathcal{B} = \bigcup_{m\in\mathcal{P}} \mathcal{B}_m.$$

Finally, we define

$$\mathcal{C} = \bigcup_{m\in\mathcal{Q}} \mathcal{G}_m.$$

and

$$\mathcal{C}_m = \mathcal{G}_m \cap \mathcal{C}.$$

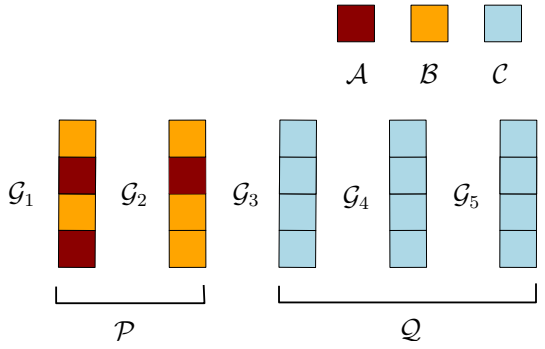

Fig. 2. Illustration of the partitioning of a 20 element coefficient vector $\mathbf{w}$ into 5 groups of 4 indices. The sets $\mathcal{P}$ and $\mathcal{Q}$ contain the active groups and the inactive groups, respectively. Within each of the two active groups the maximal coefficients are denoted by the dark red color.

B. Optimality condition

The objective function in (15) is convex but non-smooth as the $\ell_{1,\infty}$ norm is non-differentiable. Therefore, problem (15) reaches its global minimum at $\mathbf{w}$ if and only if the sub-differential of the objective function contains the zero vector. Let $\partial\|\mathbf{w}\|_{1,\infty}$ denote the sub-differential of the $\ell_{1,\infty}$ norm at $\mathbf{w}$. A vector $\mathbf{z} \in \partial\|\mathbf{w}\|_{1,\infty}$ only if $\mathbf{z}$ satisfies the following conditions (details can be found in [12], [14]):

$$\|\mathbf{z}_{\mathcal{A}_m}\|_1 = 1, m \in \mathcal{P}, \quad (16)$$

$$\operatorname{sgn}(\mathbf{z}_{\mathcal{A}_m}) = \operatorname{sgn}(\mathbf{w}_{\mathcal{A}_m}), m \in \mathcal{P}, \quad (17)$$

$$\mathbf{z}_{\mathcal{B}} = \mathbf{0}, \quad (18)$$

$$\|\mathbf{z}_{\mathcal{C}_m}\|_1 \le 1, m \in \mathcal{Q}, \quad (19)$$

where $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{P}$ and $\mathcal{Q}$ are β-dependent sets defined on $\mathbf{w}$ as defined in Section III-A.

For notational convenience we drop β in $\mathbf{R}(\beta)$ and $\mathbf{r}(\beta)$ leaving the β-dependency implicit. The optimality condition is then written as

$$\mathbf{R}\mathbf{w} - \mathbf{r} + \lambda\mathbf{z} = \mathbf{0}, \quad \mathbf{z} \in \partial\|\mathbf{w}\|_{\mathbf{1},\infty}. \quad (20)$$

As $\mathbf{w}_{\mathcal{C}} = \mathbf{0}$ and $\mathbf{z}_{\mathcal{B}} = \mathbf{0}$, (20) implies the three conditions

$$\mathbf{R}_{\mathcal{A}\mathcal{A}}\mathbf{w}_{\mathcal{A}} + \mathbf{R}_{\mathcal{A}\mathcal{B}}\mathbf{w}_{\mathcal{B}} - \mathbf{r}_{\mathcal{A}} + \lambda\mathbf{z}_{\mathcal{A}} = \mathbf{0}, \quad (21)$$

$$\mathbf{R}_{\mathcal{B}\mathcal{A}}\mathbf{w}_{\mathcal{A}} + \mathbf{R}_{\mathcal{B}\mathcal{B}}\mathbf{w}_{\mathcal{B}} - \mathbf{r}_{\mathcal{B}} = \mathbf{0}, \quad (22)$$

$$\mathbf{R}_{\mathcal{C}\mathcal{A}}\mathbf{w}_{\mathcal{A}} + \mathbf{R}_{\mathcal{C}\mathcal{B}}\mathbf{w}_{\mathcal{B}} - \mathbf{r}_{\mathcal{C}} + \lambda\mathbf{z}_{\mathcal{C}} = \mathbf{0}. \quad (23)$$

The vector $\mathbf{w}_{\mathcal{A}}$ lies in a low dimensional subspace. Indeed, by definition of $\mathcal{A}_m$, if $|\mathcal{A}_m| > 1$

$$|w_i| = |w_{i'}|, \quad i, i' \in \mathcal{A}_m.$$

Therefore, for any active group $m \in \mathcal{P}$,

$$\mathbf{w}_{\mathcal{A}_m} = \mathbf{s}_{\mathcal{A}_m}\alpha_m \quad (24)$$

where

$$\alpha_m = \|\mathbf{w}_{\mathcal{G}_m}\|_\infty,$$

and

$$\mathbf{s}_{\mathcal{A}} = \operatorname{sgn}(\mathbf{w}_{\mathcal{A}}).$$

Using matrix notation, we represent (24) as

$$\mathbf{w}_{\mathcal{A}} = \mathbf{S}\mathbf{a}. \quad (25)$$

where

$$\mathbf{S} = \begin{pmatrix} \mathbf{s}_{A_1} & & \\ & \ddots & \\ & & \mathbf{s}_{A_{|P|}} \end{pmatrix} \tag{26}$$

is a $|\mathcal{A}| \times |\mathcal{P}|$ sign matrix and the vector $\mathbf{a}$ is comprised of $\alpha_m, m \in \mathcal{P}$.

The solution to (15) can be determined in closed form if the sign matrix $\mathbf{S}$ and sets $(\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{P}, \mathcal{Q})$ are available. Indeed, from (16) and (17)

$$\mathbf{S}^T \mathbf{z}_{\mathcal{A}} = \mathbf{1}, \tag{27}$$

where $\mathbf{1}$ is a $|\mathcal{P}| \times 1$ vector comprised of 1's. With (25) and (27), (21) and (22) are equivalent to

$$\begin{aligned} &\mathbf{S}^T \mathbf{R}_{\mathcal{A}\mathcal{A}} \mathbf{S}\mathbf{a} + \mathbf{S}^T \mathbf{R}_{\mathcal{A}\mathcal{B}} \mathbf{w}_{\mathcal{B}} - \mathbf{S}^T \mathbf{r}_{\mathcal{A}} + \lambda \mathbf{1} = 0, \\ &\mathbf{R}_{\mathcal{B}\mathcal{A}} \mathbf{S}\mathbf{a} + \mathbf{R}_{\mathcal{B}\mathcal{B}} \mathbf{w}_{\mathcal{B}} - \mathbf{r}_{\mathcal{B}} = \mathbf{0}. \end{aligned} \tag{28}$$

Therefore, by defining the (a.s. invertible) matrix

$$\mathbf{H} = \begin{pmatrix} \mathbf{S}^T \mathbf{R}_{\mathcal{A}\mathcal{A}} \mathbf{S} & \mathbf{S}^T \mathbf{R}_{\mathcal{A}\mathcal{B}} \\ \mathbf{R}_{\mathcal{B}\mathcal{A}} \mathbf{S} & \mathbf{R}_{\mathcal{B}\mathcal{B}} \end{pmatrix}, \tag{29}$$

and

$$\mathbf{b} = \begin{pmatrix} \mathbf{S}^T \mathbf{r}_{\mathcal{A}} \\ \mathbf{r}_{\mathcal{B}} \end{pmatrix}, \mathbf{v} = \begin{pmatrix} \mathbf{a} \\ \mathbf{w}_B \end{pmatrix}, \tag{30}$$

(28) is equivalent to $\mathbf{H}\mathbf{v} = \mathbf{b} - \lambda \mathbf{e}$, where $\mathbf{e} = (\mathbf{1}^T, \mathbf{0}^T)^T$, so that

$$\mathbf{v} = \mathbf{H}^{-1}(\mathbf{b} - \lambda \mathbf{e}). \tag{31}$$

As $\mathbf{w}_{\mathcal{A}} = \mathbf{S}\mathbf{a}$, the solution vector $\mathbf{w}$ can be directly obtained from $\mathbf{v}$ via (30). For the sub-gradient vector, it can be shown that

$$\lambda \mathbf{z}_{\mathcal{A}} = \mathbf{r}_{\mathcal{A}} - (\mathbf{R}_{\mathcal{A}\mathcal{A}} \mathbf{S} \quad \mathbf{R}_{\mathcal{A}\mathcal{B}})\, \mathbf{v}, \tag{32}$$

$$\mathbf{z}_{\mathcal{B}} = \mathbf{0} \tag{33}$$

and

$$\lambda \mathbf{z}_{\mathcal{C}} = \mathbf{r}_{\mathcal{C}} - (\mathbf{R}_{\mathcal{C}\mathcal{A}} \mathbf{S} \quad \mathbf{R}_{\mathcal{C}\mathcal{B}})\, \mathbf{v}. \tag{34}$$

C. Online update

Now we consider (15) using the results in III-B. Let β_0 and β_1 be two constants such that $\beta_1 > \beta_0$. For a given value of $\beta \in [\beta_0, \beta_1]$ define the class of sets $\mathcal{S} = (\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{P}, \mathcal{Q})$ and make β explicit by writing $\mathcal{S}(\beta)$. Recall that $\mathcal{S}(\beta)$ is specified by the solution $f(\beta, \lambda)$ defined in (19). Assume that $\mathcal{S}(\beta)$ does not change for $\beta \in [\beta_0, \beta_1]$. The following theorem propagates $f(\beta_0, \lambda)$ to $f(\beta_1, \lambda)$ via a simple algebraic relation.

Theorem 1. *Let β_0 and β_1 be two constants such that $\beta_1 > \beta_0$ and for any $\beta \in [\beta_0, \beta_1]$ the solutions to (15) share the same sets $\mathcal{S} = (\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{P}, \mathcal{Q})$. Let $\mathbf{v}'$ and $\mathbf{v}$ be vectors defined as $f(\beta_1, \lambda)$ and $f(\beta_0, \lambda)$, respectively. Then*

$$\mathbf{v}' = \mathbf{v} + \frac{\beta_1 - \beta_0}{1 + \sigma_H^2 \beta_1} (y - \hat{y}) \mathbf{g}, \tag{35}$$

and the corresponding sub-gradient vector has the explicit update

$$\lambda \mathbf{z}'_{\mathcal{A}} = \lambda \mathbf{z}_{\mathcal{A}} + \frac{\beta_1 - \beta_0}{1 + \sigma_H^2 \beta_1} (y - \hat{y}) \left\{ \mathbf{x}_{\mathcal{A}} - (\mathbf{R}_{\mathcal{A}\mathcal{A}} \mathbf{S} \quad \mathbf{R}_{\mathcal{A}\mathcal{B}}) \mathbf{g} \right\} \tag{36}$$

and

$$\lambda \mathbf{z}'_{\mathcal{C}} = \lambda \mathbf{z}_{\mathcal{C}} + \frac{\beta_1 - \beta_0}{1 + \sigma_H^2 \beta_1} (y - \hat{y}) \left\{ \mathbf{x}_{\mathcal{C}} - (\mathbf{R}_{\mathcal{C}\mathcal{A}} \mathbf{S} \quad \mathbf{R}_{\mathcal{C}\mathcal{B}}) \mathbf{g} \right\}, \tag{37}$$

where $\mathbf{R} = \mathbf{R}(0)$ as defined in (13), $(\mathbf{x}, y)$ is the new sample as defined in (13) and (14), the sign matrix $\mathbf{S}$ is obtained from the solution at $\beta = \beta_0$, $\mathbf{H}_0$ is calculated from (29) using $\mathbf{S}$ and $\mathbf{R}(0)$, and $\mathbf{d}$, $\mathbf{u}$, $\hat{y}$ and σ_H^2 are defined by

$$\mathbf{d} = \begin{pmatrix} \mathbf{S}^T \mathbf{x}_{\mathcal{A}} \\ \mathbf{x}_{\mathcal{B}} \end{pmatrix}, \tag{38}$$

$$\mathbf{g} = \mathbf{H}_0^{-1} \mathbf{d}, \tag{39}$$

$$\hat{y} = \mathbf{d}^T \mathbf{v}, \tag{40}$$

$$\sigma_H^2 = \mathbf{d}^T \mathbf{g}. \tag{41}$$

The proof of Theorem 1 is provided in Appendix A. Theorem 1 provides the closed form update for the solution path $f(\beta_0, \lambda) \rightarrow f(\beta_1, \lambda)$, under the assumption that the associated sets $\mathcal{S}(\beta)$ remain unaltered over the path.

Next, we partition the range $\beta \in [0, 1]$ into contiguous segments over which $\mathbf{S}(\beta)$ is piecewise constant. Within each segment we can use Theorem 1 to propagate the solution from left endpoint to right endpoint. Below we specify an algorithm for finding the endpoints of each of these segments.

Fix an endpoint β_0 of one of these segments. We seek a *critical point* β_1 that is defined as the maximum β_1 ensuring $\mathcal{S}(\beta)$ remains unchanged within $[\beta_0, \beta_1]$. By increasing β_1 from β_0, the sets $\mathcal{S}(\beta)$ will not change until at least one of the following conditions are met:

Condition 1. There exists $i \in \mathcal{A}$ such that $z'_i = 0$;
Condition 2. There exists $i \in \mathcal{B}_m$ such that $|w'_i| = \alpha'_m$;
Condition 3. There exists $m \in \mathcal{P}$ such that $\alpha'_m = 0$;
Condition 4. There exists $m \in \mathcal{Q}$ such that $\|\mathbf{z}'_{\mathcal{C}_m}\|_1 = 1$.

Condition 1 is from (17) and (18), Condition 2 and 3 are based on definitions of $\mathcal{A}$ and $\mathcal{P}$, respectively, and Condition 4 comes from (16) and (19). Following [12], [22], the four conditions can be assumed to be mutually exclusive. The actions with respect to Conditions 1-4 are given by

Action 1. Move the entry i from $\mathcal{A}$ to $\mathcal{B}$:

$$\mathcal{A} \leftarrow \mathcal{A} - \{i\}, \mathcal{B} \leftarrow \mathcal{B} \cup \{i\};$$

Action 2. Move the entry i from $\mathcal{B}$ to $\mathcal{A}$:

$$\mathcal{A} \leftarrow \mathcal{A} \cup \{i\}, \mathcal{B} \leftarrow \mathcal{B} - \{i\};$$

Action 3. Remove group m from the active group list

$$\mathcal{P} \leftarrow \mathcal{P} - \{m\}, \mathcal{Q} \leftarrow \mathcal{Q} \cup \{m\},$$

and update the related sets

$$\mathcal{A} \leftarrow \mathcal{A} - \mathcal{A}_m, \mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{A}_m;$$

Action 4. Select group m

$$\mathcal{P} \leftarrow \mathcal{P} \cup \{m\}, \mathcal{Q} \leftarrow \mathcal{Q} - \{m\},$$

and update the related sets

$$\mathcal{A} \leftarrow \mathcal{A} \cup \mathcal{C}_m, \mathcal{C} \leftarrow \mathcal{C} - \mathcal{C}_m.$$

By Theorem 1, the solution update from β_0 to β_1 is in closed form. The critical point of β_1 can be determined in a straightforward manner (details are provided in Appendix B). Let $\beta_1^{(k)}, k = 1, ..., 4$ be the minimum value that is greater than β_0 and meets Condition 1-4, respectively. The critical point β_1 is then

$$\beta_1 = \min_{k=1,...,4} \beta_1^{(k)}.$$

D. Homotopy algorithm implementation

We now have all the ingredients for the homotopy update algorithm and the pseudo code is given in Algorithm 1.

Algorithm 1: Homotopy update from $f(0, \lambda)$ to $f(1, \lambda)$.

Input : $f(0, \lambda), \mathbf{R}(0), \mathbf{x}, \mathbf{y}$
output: $f(1, \lambda)$

Initialize $\beta_0 = 0$, $\beta_1 = 0$, $\mathbf{R} = \mathbf{R}(0)$;
Calculate $(\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{P}, \mathcal{Q})$ and $(\mathbf{v}, \lambda \mathbf{z}_\mathcal{A}, \lambda \mathbf{z}_\mathcal{C})$ from $f(0, \lambda)$;
while $\beta_0 < 1$ **do**
  Calculate the environmental variables $(\mathbf{S}, \mathbf{H}_0, \mathbf{d}, \mathbf{g}, \hat{y}, \sigma_H^2)$ from $f(\beta_0, \lambda)$ and $\mathbf{R}$;
  Calculate $\{\beta_1^{(k)}\}_{k=1}^4$ that meets Condition 1-4, respectively;
  Calculate the critical point β_1 that meets Condition k_*: $k_* = \arg\min_k \beta_1^{(k)}$ and $\beta_1 = \beta_1^{(k_*)}$;
  if $\beta_1 \leq 1$ **then**
    Update $(\mathbf{v}, \lambda \mathbf{z}_\mathcal{A}, \lambda \mathbf{z}_\mathcal{C})$ using (35), (36) and (37);
    Update $(\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{P}, \mathcal{Q})$ by Action k_*;
    $\beta_0 = \beta_1$;
  else
    break;
  end
end
$\beta_1 = 1$;
Update $(\mathbf{v}, \lambda \mathbf{z}_\mathcal{A}, \lambda \mathbf{z}_\mathcal{C})$ using (35);
Calculate $f(1, \lambda)$ from $\mathbf{v}$.

Next we analyze the computational cost of Algorithm 1. The complexity to compute each critical point is summarized in Table I, where N is the dimension of $\mathbf{H}_0$. As $N = |\mathcal{P}| + |\mathcal{B}| \leq |\mathcal{A}| + |\mathcal{B}|$, N is upper bounded by the number of non-zeros in the solution vector. The vector $\mathbf{g}$ can be computed in $\mathcal{O}(N^2)$ time using the matrix-inverse lemma [20] and the fact that, for each action, $\mathbf{H}_0$ is at most perturbed by a rank-two matrix. This implies that the computation complexity per critical point is $\mathcal{O}(p \max\{N, \log p\})$ and the total complexity of the online update is $\mathcal{O}(k_2 \cdot p \max\{N, \log p\})$, where k_2 is the number of critical points of β in the solution path $f(0, \lambda) \to f(1, \lambda)$. This is the computational cost required for Step 2 in Section II-C.

A similar analysis can be performed for the complexity of Step 1, which requires $\mathcal{O}(k_1 \cdot p \max\{N, \log p\})$ where k_1 is the number of critical points in the solution path $f(0, \gamma\lambda) \to f(0, \lambda)$. Therefore, the overall computation complexity of the recursive $\ell_{1,\infty}$ group lasso is $\mathcal{O}(k \cdot p \max\{N, \log p\})$, where

$\mathbf{g} = \mathbf{H}_0^{-1}\mathbf{d}$	$\mathcal{O}(N^2)$
$\mathbf{x}_\mathcal{A} - (\mathbf{R}_{\mathcal{A}\mathcal{A}}\mathbf{S} \;\; \mathbf{R}_{\mathcal{A}\mathcal{B}})\mathbf{g}$	$\mathcal{O}(\lvert\mathcal{A}\rvert N)$
$\mathbf{x}_\mathcal{C} - (\mathbf{R}_{\mathcal{C}\mathcal{A}}\mathbf{S} \;\; \mathbf{R}_{\mathcal{C}\mathcal{B}})\mathbf{g}$	$\mathcal{O}(\lvert\mathcal{C}\rvert N)$
$\beta_1^{(1)}$	$\mathcal{O}(\lvert\mathcal{A}\rvert)$
$\beta_1^{(2)}$	$\mathcal{O}(\lvert\mathcal{B}\rvert)$
$\beta_1^{(3)}$	$\mathcal{O}(\lvert\mathcal{P}\rvert)$
$\beta_1^{(4)}$	$\mathcal{O}(\lvert\mathcal{C}\rvert \log \lvert\mathcal{C}\rvert)$

TABLE I
COMPUTATION COSTS OF ONLINE HOMOTOPY UPDATE FOR EACH CRITICAL POINT.

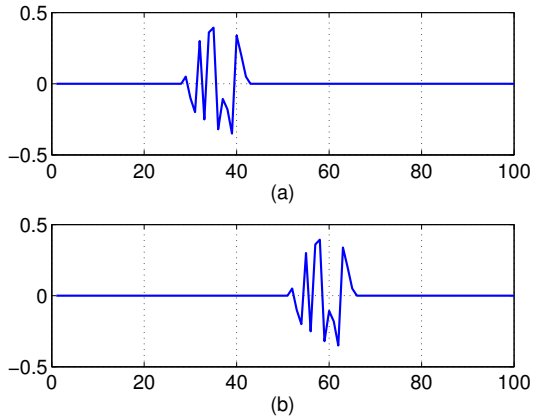

Fig. 3. Responses of the time varying system. (a): Initial response. (b): Response after the 200th iteration. The groups for Algorithm 1 were chosen as 20 equal size contiguous groups of coefficients partitioning the range $1, \ldots, 100$.

$k = k_1 + k_2$, *i.e.,* the total number of critical points in the solution path $f(0, \gamma\lambda) \to f(0, \lambda) \to f(1, \lambda)$.

An instructive benchmark is to directly solve the n-samples problem (12) from the solution path $f(1, \infty)$ (*i.e.,* a zero vector) $\to f(1, \lambda)$ [12], without using the previous solution $\hat{\mathbf{w}}_{n-1}$. This algorithm, called iCap in [12], requires $\mathcal{O}(k' \cdot p \max\{N, \log p\})$, where k' is the number of critical points in $f(1, \infty) \to f(1, \lambda)$. Empirical comparisons between k and k', provided in the following section, indicate that iCap requires significantly more computation than our proposed Algorithm 1.

IV. NUMERICAL SIMULATIONS

In this section we demonstrate our proposed recursive $\ell_{1,\infty}$ group lasso algorithm by numerical simulation. We simulated the model $y_j = \mathbf{w}_*^T \mathbf{x}_j + v_j$, $j = 1, \ldots, 400$, where v_j is a zero mean Gaussian noise and $\mathbf{w}_*$ is a sparse $p = 100$ element vector containing only 14 non-zero coefficients clustered between indices 29 and 42. See Fig. 3 (a). After 200 time units, the locations of the non-zero coefficients of $\mathbf{w}_*$ is shifted to the right, as indicated in Fig. 3 (b).

The input vectors were generated as independent identically distributed Gaussian random vectors with zero mean and identity covariance matrix, and the variance of observation noise v_j is 0.01. We created the groups in the recursive $\ell_{1,\infty}$ group lasso as follows. We divide the 100 RLS filter coefficients $\mathbf{w}$ into 20 groups with group boundaries $1, 5, 10, \ldots$, where each group contains 5 coefficients. The forgetting factor γ

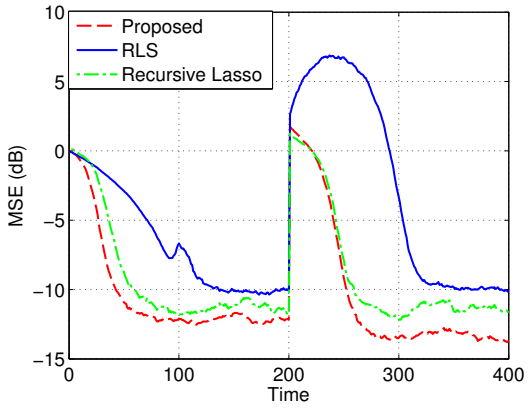

Fig. 4. Averaged MSE of the proposed algorithm, RLS and recursive lasso. The corresponding system response is shown in Fig. 3.

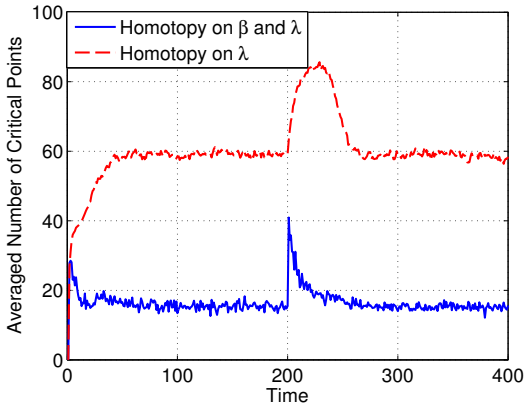

Fig. 5. Averaged number of critical points for the proposed recursive method of implementing $\ell_{1,\infty}$ lasso and the iCap [12] non-recursive method of implementation. The corresponding system response is shown in Fig. 3.

and the regularization parameter λ were set to 0.9 and 0.1, respectively. We repeated the simulation 100 times and the averaged mean squared errors of the RLS, sparse RLS and proposed RLS shown in Fig. 4. We implemented the standard RLS and sparse RLS using the ℓ_1 regularization, where the forgetting factors are also set to 0.9. As [9], [10] and [15] are solving the same online optimization problem it suffices to compare the MSE of [15] to that of sparse RLS. The regularization parameter λ was chosen to achieve the lowest steady-state error, resulting in $\lambda = 0.05$. It is important to note that, while their computational complexity is lower than that of our proposed RLS algorithm (and lower than that of [15]), the methods of [9] and [10] only approximate the exact ℓ_1 penalized solution.

It can be seen from Fig. 4 that our proposed sparse RLS method outperforms standard RLS and sparse RLS in both convergence rate and steady-state MSE. This demonstrates the power of our group sparsity penalty. At the change point of 200 iterations, both the proposed method and sparse RLS of [15] show superior tracking performances as compared to the standard RLS. We also observe that the proposed method achieves even smaller MSE after the change point occurs. This is due to the fact that the active cluster spans across group boundaries in the initial system (Fig. 3 (a)), while the active clusters in the shifted system overlap with fewer groups.

Fig. 5 shows the average number of critical points (accounting for both trajectories in β and λ) of the proposed algorithm, *i.e.,* the number k as defined in Section III-D. As a comparison, we implement the iCap method of [12], a homotopy based algorithm that traces the solution path only over λ. The average number of critical points for iCap is plotted in Fig. 5, which is the number k' in Section III-D. Both the proposed algorithm and iCap yield the same solution but have different computational complexities proportional to k and k', respectively. It can be seen that the proposed algorithm saves as much as 75% of the computation costs for equivalent performance.

We next investigated the effect of adding a second group of non-zero coefficients to $\mathbf{w}_*$ and compare different algorithms for group-sparse RLS algorithms. Fig. 6(a)-(b) shows these 100 coefficients before and after a relocation of both the two groups in reverse directions at the 200th time point. The parameter settings and group allocation setting for our RLS algorithm are the same as before. The regularization parameter λ is set to 0.03 for the proposed method. For comparison, we utilized the SpaRSA algorithm [19] implemented in a public software package[1] to solve group-sparse regularized RLS problems in an online manner. The SpaRSA algorithm is an efficient iterative method in which each step is obtained by solving an optimization subproblem, and it supports warm-start capabilities by choosing good initial values. In our online implementation, the current estimate is used as the initial value for solving the next estimate using the updated measurement sample. Both $\ell_{1,\infty}$ and $\ell_{1,2}$ regularizers are employed in the SpaRSA algorithm. The regularization parameter λ of $\ell_{1,\infty}$-SpaRSA is set to the same value as in the proposed method and the λ for $\ell_{1,2}$-SpaRSA is tuned for best steady-state MSE. As SpaRSA is an iterative algorithm, its solution is dependent on the stopping rule, which is selected based on stopping threshold on the relative change in the objective value. Two stopping threshold values, 10^{-3} and 10^{-5}, are set for $\ell_{1,\infty}$-SpaRSA, respectively, and 10^{-5} is set to $\ell_{1,2}$-SpaRSA,

The MSE performance comparison is shown in Fig. 7 and the averaged computational time performed on an Intel Core 2.53GHz CPU is plotted in Fig. 8. It can be observed from Fig. 7 and Fig. 8 that both the MSE performance and computational time of SpaRSA algorithms depend on the stopping threshold. This is due to the fact that SpaRSA obtains approximate solutions of the regularized RLS problem rather than the exact solution attained by the proposed method. For $\ell_{1,\infty}$-SpaRSA, a large threshold value implies faster run time but less tracking performance in MSE as compared to a smaller stopping threshold. We observed that the MSE curve of the more computationally demanding $\ell_{1,\infty}$-SpaRSA will approach to that of the proposed method when the stopping threshold is very large as the two algorithms solve the same optimization problem (results not shown here). For $\ell_{1,2}$-SpaRSA, the algorithm achieves comparable computational time with the proposed method with the specified stopping threshold, while

[1]Available from http://www.lx.it.pt/~mtf/SpaRSA/

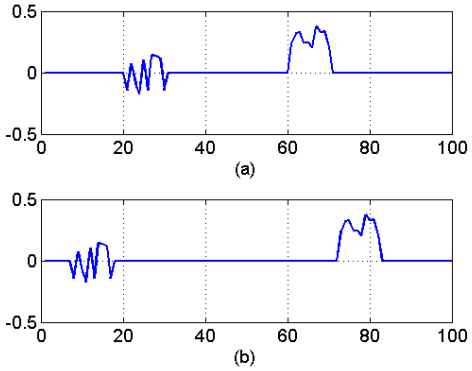

Fig. 6. Coefficients for the spurious group misalignment experiment shown in Figs. 7 and 8. (a): Initial response. (b): Response after the 200th iteration. The groups for the recursive group sparse RLS algorithm (Algorithm 1) were chosen as 20 equal size contiguous groups of coefficients partitioning the range $1, \ldots, 100$.

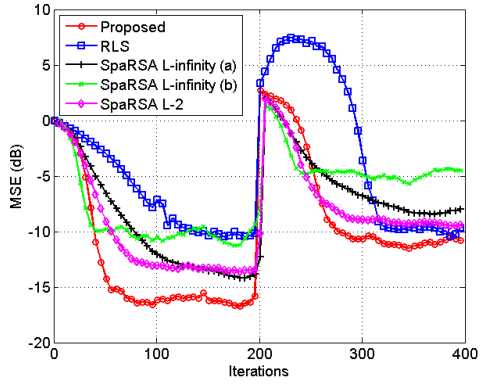

Fig. 7. Averaged MSE of the proposed algorithm, RLS and SpaRSA algorithms [19]. The stopping thresholds for $\ell_{1,\infty}$-SpaRSA (a), $\ell_{1,\infty}$-SpaRSA (b) and $\ell_{1,2}$-SpaRSA are set to 10^{-3}, 10^{-5} and 10^{-5}, respectively. The corresponding system response is shown in Fig. 6.

the MSE performance of $\ell_{1,2}$-SpaRSA is worse than that of the proposed method.

Note that the system response before the 200th iteration is perfectly aligned with the group allocation but is shifted out of alignment after the change point. The difference between the MSE curves of the proposed method before and after the transition suggests that our proposed recursive group lasso is not overly sensitive to shifts in the group-partition even when there are multiple groups.

V. Conclusion

In this paper we proposed a $\ell_{1,\infty}$ regularized RLS algorithm for online sparse linear prediction. We developed a homotopy based method to sequentially update the solution vector as new measurements are acquired. Our proposed algorithm uses the previous estimate as a "warm-start", from which we compute the homotopy update to the exact current solution. The proposed algorithm can process streaming measurements with time varying predictors and is computationally efficient compared to non-recursive and contemporary warm-start capable group lasso solvers. Numerical simulations demonstrated

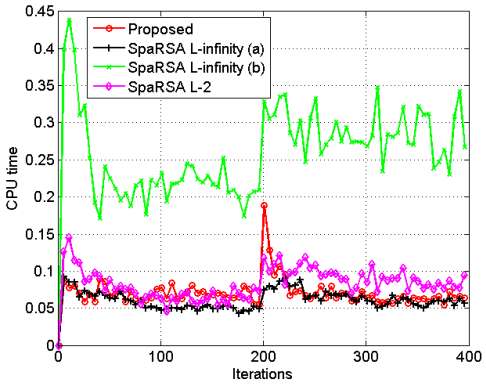

Fig. 8. Averaged computational time of the proposed algorithm, RLS and SpaRSA algorithms [19] using an Intel Core 2.53GHz CPU. The stopping thresholds for $\ell_{1,\infty}$-SpaRSA (a), $\ell_{1,\infty}$-SpaRSA (b) and $\ell_{1,2}$-SpaRSA are set to 10^{-3}, 10^{-5} and 10^{-5}, respectively. The corresponding system response is shown in Fig. 6.

that the proposed method outperformed the standard and ℓ_1 regularized RLS for identifying an unknown group sparse system, in terms of both tracking and steady-state mean squared error.

As indicated in Section IV, the MSE performance of the proposed method depends on the choice of group partition. The development of performance-optimal data-dependent group partitioning algorithms is a worthwhile topic for future study. Another worthwhile area of research is the extension of the proposed method to overlapping groups and other flexible partitions [23].

VI. Appendix

The authors would like to thank the reviewers for their valuable comments on the paper.

VII. Appendix

A. Proof of Theorem 1

We begin by deriving (35). According to (31),

$$\mathbf{v}' = \mathbf{H}'^{-1}(\mathbf{b}' - \lambda \mathbf{e}'). \tag{42}$$

As $\mathbf{S}$ and $(\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{P}, \mathcal{Q})$ remain constant within $[\beta_0, \beta_1]$,

$$\mathbf{e}' = \mathbf{e}, \tag{43}$$

$$\mathbf{b}' = \mathbf{b} + \delta \mathbf{d} y, \tag{44}$$

and

$$\mathbf{H}' = \mathbf{H} + \delta \mathbf{d}\mathbf{d}^T,$$

where

$$\delta = \beta_1 - \beta_0,$$

$\mathbf{H}$ and $\mathbf{b}$ are calculated using $\mathbf{S}$ within $[\beta_0, \beta_1]$ and $\mathbf{R}(\beta_0)$ and $\mathbf{r}(\beta_0)$, respectively. We emphasize that $\mathbf{H}$ is based on $\mathbf{R}(\beta)$ and is different from $\mathbf{H}_0$ defined in Theorem 1. According to the Sherman-Morrison-Woodbury formula,

$$\mathbf{H}'^{-1} = \mathbf{H}^{-1} - \frac{\delta}{1 + \sigma^2 \delta}(\mathbf{H}^{-1}\mathbf{d})(\mathbf{H}^{-1}\mathbf{d})^T, \tag{45}$$

where $\sigma^2 = \mathbf{d}^T\mathbf{H}^{-1}\mathbf{d}$. Substituting (43), (44) and (45) into (42), after simplification we obtain

$$\begin{aligned}\mathbf{v}' &= \left(\mathbf{H}^{-1} - \frac{\delta}{1+\sigma^2\delta}(\mathbf{H}^{-1}\mathbf{d})(\mathbf{H}^{-1}\mathbf{d})^T\right)(\mathbf{b} + \delta\mathbf{d}y - \lambda\mathbf{e}) \\ &= \mathbf{H}^{-1}(\mathbf{b} - \lambda\mathbf{e}) + \mathbf{H}^{-1}\delta\mathbf{d}y \\ &\quad - \frac{\delta}{1+\sigma^2\delta}\mathbf{H}^{-1}\mathbf{d}\mathbf{d}^T\mathbf{H}^{-1}(\mathbf{b} - \lambda\mathbf{e}) - \frac{\sigma^2\delta^2}{1+\sigma^2\delta}\mathbf{H}^{-1}\mathbf{d}y \\ &= \mathbf{v} + \frac{\delta}{1+\sigma^2\delta}(y - \mathbf{d}^T\mathbf{v})\mathbf{H}^{-1}\mathbf{d} \\ &= \mathbf{v} + \frac{\delta}{1+\sigma^2\delta}(y - \hat{y})\mathbf{H}^{-1}\mathbf{d},\end{aligned} \tag{46}$$

where $\hat{y} = \mathbf{d}^T\mathbf{v}$ as defined in (40).

Note that $\mathbf{H}$ is defined in terms of $\mathbf{R}(\beta_0)$ rather than $\mathbf{R}(0)$ and

$$\mathbf{H} = \mathbf{H}_0 + \beta_0\mathbf{d}\mathbf{d}^T,$$

so that

$$\mathbf{H}^{-1} = \mathbf{H}_0^{-1} - \frac{\beta_0}{1+\sigma_H^2\beta_0}\mathbf{g}\mathbf{g}^T, \tag{47}$$

where $\mathbf{g}$ and σ_H^2 are defined by (39) and (41), respectively. As $\sigma_H^2 = \mathbf{d}^T\mathbf{g}$,

$$\mathbf{H}^{-1}\mathbf{d} = \mathbf{H}_0^{-1}\mathbf{d} - \frac{\sigma_H^2\beta_0}{1+\sigma_H^2\beta_0}\mathbf{g}. \tag{48}$$

Accordingly,

$$\sigma^2 = \mathbf{d}^T\mathbf{H}^{-1}\mathbf{d} = \sigma_H^2 - \frac{\sigma_H^2\beta_0}{1+\sigma_H^2\beta_0}\sigma_H^2 = \frac{\sigma_H^2}{1+\sigma_H^2\beta_0}. \tag{49}$$

Substituting (48) and (49) to (46), we finally obtain

$$\mathbf{v}' = \mathbf{v} + \frac{\delta}{1+\sigma_H^2\beta_1}(y - \hat{y})\mathbf{g} = \mathbf{v} + \frac{\beta_1 - \beta_0}{1+\sigma_H^2\beta_1}(y - \hat{y})\mathbf{g}.$$

Equations (36) and (37) can be established by direct substitutions of (35) into their definitions (32) and (34) and thus the proof of Theorem 1 is complete.

B. Computation of critical points

For ease of notation we work with ρ, defined by

$$\rho = \frac{\beta_1 - \beta_0}{1+\sigma_H^2\beta_1}. \tag{50}$$

It is easy to see that over the range $\beta_1 > \beta_0$, ρ is monotonically increasing in $(0, 1/\sigma_H^2)$. Therefore, (50) can be inverted by

$$\beta_1 = \frac{\rho + \beta_0}{1-\sigma_H^2\rho}, \tag{51}$$

where $\rho \in (0, 1/\sigma_H^2)$ to ensure $\beta_1 > \beta_0$.

Suppose we have obtained $\rho^{(k)}, k = 1, ..., 4$, $\beta_1^{(k)}$ can be calculated using (51) and the critical point β_1 is then

$$\beta_1 = \min_{k=1,...,4}\beta_1^{(k)}.$$

We now calculate the critical value of ρ for each condition one by one.

1) Critical point for Condition 1: Define the temporary vector

$$\mathbf{t}_{\mathcal{A}} = (y - \hat{y})\left\{\mathbf{x}_{\mathcal{A}} - (\mathbf{R}_{\mathcal{A}\mathcal{A}}\mathbf{S} \;\; \mathbf{R}_{\mathcal{A}\mathcal{B}})\,\mathbf{g}\right\}.$$

According to (36),

$$\lambda\mathbf{z}'_{\mathcal{A}} = \lambda\mathbf{z}_{\mathcal{A}} + \rho\mathbf{t}_{\mathcal{A}}.$$

Condition 1 is met for any $\rho = \rho_i^{(1)}$ such that

$$\rho_i^{(1)} = -\frac{\lambda z_i}{t_i}, i \in \mathcal{A}.$$

Therefore, the critical value of ρ that satisfies Condition 1 is

$$\rho^{(1)} = \min\left\{\rho_i^{(1)} \,\middle|\, i \in \mathcal{A}, \rho_i^{(1)} \in (0, 1/\sigma_H^2)\right\}.$$

2) Critical point for Condition 2: By the definition (30), $\mathbf{v}$ is a concatenation of α_m and $\mathbf{w}_{\mathcal{B}_m}, m \in \mathcal{P}$:

$$\mathbf{v}^T = \left((\alpha_m)_{m\in\mathcal{P}}, \mathbf{w}_{\mathcal{B}_1}^T, ..., \mathbf{w}_{\mathcal{B}_{|\mathcal{P}|}}^T\right), \tag{52}$$

where $(\alpha_m)_{m\in\mathcal{P}}$ denotes the vector comprised of $\alpha_m, m \in \mathcal{P}$. Now we partition the vector $\mathbf{g}$ in the same manner as (52) and denote τ_m and $\mathbf{u}_m$ as the counter part of α_m and $\mathbf{w}_{\mathcal{B}_m}$ in $\mathbf{g}$, *i.e.,*

$$\mathbf{g}^T = \left((\tau_m)_{m\in\mathcal{P}}, \mathbf{u}_1^T, ..., \mathbf{u}_{|\mathcal{P}|}^T\right).$$

Eq. (35) is then equivalent to

$$\alpha'_m = \alpha_m + \rho\tau_m, \tag{53}$$

and

$$w'_{\mathcal{B}_m,i} = w_{\mathcal{B}_m,i} + \rho u_{m,i},$$

where $u_{m.i}$ is the i-th element of the vector $\mathbf{u}_m$. Condition 2 indicates that

$$\alpha'_m = \pm w'_{\mathcal{B}_m,i},$$

and is satisfied if $\rho = \rho_{m,i}^{(2+)}$ or $\rho = \rho_{m,i}^{(2-)}$, where

$$\rho_{m,i}^{(2+)} = \frac{\alpha_m - w_{\mathcal{B}_m,i}}{u_{m,i} - \tau_m}, \quad \rho_{m,i}^{(2-)} = -\frac{\alpha_m + w_{\mathcal{B}_m,i}}{u_{m,i} + \tau_m}.$$

Therefore, the critical value of ρ for Condition 2 is

$$\rho^{(2)} = \min\left\{\rho_{m,i}^{(2\pm)} \,\middle|\, m \in \mathcal{P}, i = 1, ..., |\mathcal{B}_m|, \rho_{m,i}^{(2\pm)} \in (0, 1/\sigma_H^2)\right\}.$$

3) Critical point for Condition 3: According to (53), $\alpha'_m = 0$ yields $\rho = \rho_i^{(3)}$ determined by

$$\rho_m^{(3)} = -\frac{\alpha_m}{\tau_m}, m \in \mathcal{P},$$

and the critical value for $\rho^{(3)}$ is

$$\rho^{(3)} = \min\left\{\rho_m^{(3)} \,\middle|, m \in \mathcal{P}, \rho_m^{(3)} \in (0, 1/\sigma_H^2)\right\}.$$

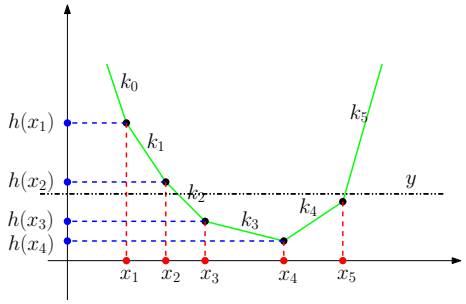

Fig. 9. An illustration of the fast algorithm for critical condition 4.

4) Critical point for Condition 4: Define

$$\mathbf{t}_{\mathcal{C}} = (y - \hat{y}) \left\{ \mathbf{x}_{\mathcal{C}} - (\mathbf{R}_{\mathcal{C}\mathcal{A}}\mathbf{S} \quad \mathbf{R}_{\mathcal{C}\mathcal{B}})\, \mathbf{g} \right\}.$$

Eq. (37) is then

$$\lambda \mathbf{z}'_{\mathcal{C}_m} = \lambda \mathbf{z}_{\mathcal{C}_m} + \rho \mathbf{t}_{\mathcal{C}_m},$$

and Condition 4 is equivalent to

$$\sum_{i \in \mathcal{C}_m} |\rho t_i + \lambda z_i| = \lambda. \tag{54}$$

To solve (54) we develop a fast method that requires complexity of $\mathcal{O}(N \log N)$, where $N = |\mathcal{C}_m|$. The algorithm is given in Appendix C. For each $m \in \mathcal{Q}$, let $\rho_m^{(4)}$ be the minimum positive solution to (54). The critical value of ρ for Condition 4 is then

$$\rho^{(4)} = \min \left\{ \rho_m^{(4)} \,\middle|\, m \in \mathcal{Q}, \rho_m^{(4)} \in (0, 1/\sigma_H^2) \right\}.$$

C. Fast algorithm for critical condition 4

Here we develop an algorithm to solve problem (54). Consider solving the more general problem:

$$\sum_{i=1}^{N} a_i |x - x_i| = y, \tag{55}$$

where a_i and x_i are constants and $a_i > 0$. Please note that the notations here have no connections to those in previous sections. Define the following function

$$h(x) = \sum_{i=1}^{N} a_i |x - x_i|.$$

The problem is then equivalent to finding $h^{-1}(y)$, if it exists.

An illustration of the function $h(x)$ is shown in Fig. 9, where k_i denotes the slope of the ith segment. It can be shown that $h(x)$ is piecewise linear and convex in x. Therefore, the equation (55) generally has two solutions if they exist, denoted as $x_{\min}$ and $x_{\max}$. Based on piecewise linearity we propose a search algorithm to solve (55). The pseudo code is shown in Algorithm 2 and its computation complexity is dominated by the sorting operation which requires $\mathcal{O}(N \log N)$.

Algorithm 2: Solve x from $\sum_{i=1}^{N} a_i |x - x_i| = y$.

Input : $\{a_i, x_i\}_{i=1}^N$, y
output: $x_{\min}, x_{\max}$

Sort $\{x_i\}_{i=1}^N$ in the ascending order: $x_1 \le x_2 \le ... \le x_N$;
Re-order $\{a_i\}_{i=1}^N$ such that a_i corresponds to x_i;
Set $k_0 = -\sum_{i=1}^N a_i$;
for $i = 1, ..., N$ **do**
  $k_i = k_{i-1} + 2a_i$;
end
Calculate $h_1 = \sum_{i=2}^N a_i |x_1 - x_i|$;
for $i = 2, ..., N$ **do**
  $h_i = h_{i-1} + k_{i-1}(x_i - x_{i-1})$
end
if $\min_i k_i > y$ **then**
  No solution;
  Exit;
else
  if $y > h_1$ **then**
    $x_{\min} = x_1 + (y - h_1)/k_0$;
  else
    Seek j such that $y \in [h_j, h_{j-1}]$;
    $x_{\min} = x_j + (y - h_j)/k_{j-1}$;
  end
  if $y > h_N$ **then**
    $x_{\max} = x_N + (y - h_N)/k_N$;
  else
    Seek j such that $y \in [h_{j-1}, h_j]$;
    $x_{\max} = x_{j-1} + (y - h_{j-1})/k_{j-1}$;
  end
end

REFERENCES

[1] Y.C. Eldar, P. Kuppinger, and H. Bolcskei, "Block-sparse signals: Uncertainty relations and efficient recovery," *Signal Processing, IEEE Transactions on*, vol. 58, no. 6, pp. 3042–3054, 2010.

[2] Y. Chen, Y. Gu, and A.O. Hero, "Regularized Least-Mean-Square Algorithms," *Arxiv preprint arXiv:1012.5066*, 2010.

[3] W.F. Schreiber, "Advanced television systems for terrestrial broadcasting: Some problems and some proposed solutions," *Proceedings of the IEEE*, vol. 83, no. 6, pp. 958–981, 1995.

[4] Y. Gu, J. Jin, and S. Mei, "ℓ_0 Norm Constraint LMS Algorithm for Sparse System Identification," *IEEE Signal Processing Letters*, vol. 16, pp. 774–777, 2009.

[5] M. Mishali and Y.C. Eldar, "From theory to practice: Sub-Nyquist sampling of sparse wideband analog signals," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 4, no. 2, pp. 375–391, 2010.

[6] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Royal. Statist. Soc B.*, vol. 58, pp. 267–288, 1996.

[7] E. Candès, "Compressive sampling," *Int. Congress of Mathematics*, vol. 3, pp. 1433–1452, 2006.

[8] Y. Chen, Y. Gu, and A.O. Hero, "Sparse LMS for system identification," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 3125–3128.

[9] B. Babadi, N. Kalouptsidis, and V. Tarokh, "SPARLS: The sparse RLS algorithm," *Signal Processing, IEEE Transactions on*, vol. 58, no. 8, pp. 4013–4025, 2010.

[10] D. Angelosante, J.A. Bazerque, and G.B. Giannakis, "Online Adaptive Estimation of Sparse Signals: Where RLS Meets the ℓ_1 norm," *Signal Processing, IEEE Transactions on*, vol. 58, no. 7, pp. 3436–3447, 2010.

[11] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[12] P. Zhao, G. Rocha, and B. Yu, "The composite absolute penalties family for grouped and hierarchical variable selection," *Annals of Statistics*, vol. 37, no. 6A, pp. 3468–3497, 2009.

[13] F.R. Bach, "Consistency of the group Lasso and multiple kernel learning," *The Journal of Machine Learning Research*, vol. 9, pp. 1179–1225, 2008.

[14] S. Negahban and M.J. Wainwright, "Joint support recovery under high-dimensional scaling: Benefits and perils of $\ell_{1,\infty}$-regularization," *Advances in Neural Information Processing Systems*, pp. 1161–1168, 2008.

[15] P.J. Garrigues and E.L. Ghaoui, "An homotopy algorithm for the Lasso with online observations," in *Neural Information Processing Systems (NIPS)*, 2008, vol. 21.

[16] S. Asif and J. Romberg, "Dynamic Updating for ℓ_1 Minimization," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 4, no. 2, pp. 421–434, 2010.

[17] D.M. Malioutov, S.R. Sanghavi, and A.S. Willsky, "Sequential compressed sensing," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 4, no. 2, pp. 435–444, 2010.

[18] S.N. Negahban and M.J. Wainwright, "Simultaneous support recovery in high dimensions: Benefits and perils of block $\ell_1\ell_\infty$-regularization," *Information Theory, IEEE Transactions on*, vol. 57, no. 6, pp. 3841–3863, 2011.

[19] S.J. Wright, R.D. Nowak, and M.A.T. Figueiredo, "Sparse reconstruction by separable approximation," *Signal Processing, IEEE Transactions on*, vol. 57, no. 7, pp. 2479–2493, 2009.

[20] W.W. Hager, "Updating the inverse of a matrix," *SIAM review*, vol. 31, no. 2, pp. 221–239, 1989.

[21] B. Widrow and S.D. Stearns, *Adaptive Signal Processing*, New Jersey: Prentice Hall, 1985.

[22] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Annals of statistics*, vol. 32, no. 2, pp. 407–451, 2004.

[23] R. Jenatton, J.Y. Audibert, and F. Bach, "Structured Variable Selection with Sparsity-Inducing Norms," *Arxiv preprint arXiv:0904.3523*, 2009.